Kiss Zoltán - Endrich Bauelemente Vertriebs GmbH

GigaDevice 32 bit ARM® Cortex®-mikrokontrollerek (3. rész)

A cikksorozat első részében áttekintettük a GigaDevice GD32™ARM® Cortex® RISC MCU sorozat architektúráját, később ismertetésre került a mikrokontroller családhoz kapható kiértékelő és kezdőkészlet is. Ebben az írásban bemutatjuk azt, hogy miként lehet ezekkel az eszközökkel megkezdeni a munkát a CrossStudio for ARM 4.1. fejlesztőrendszer használatával és példaként megírunk néhány C++ mintaprogramot, melyek az MCU egyes részeinek működtetését végzik és programozzák a GPIO és az ADC portokat.

A GD32® egy új, ARM® Cortex®-M3 vagy Cortex®-M4 32 bites RISC magokkal ellátott, alacsony fogyasztású, univerzális, nagy teljesítményű mikro vezérlő család, mely integrálja a tervezés egyszerűsítéséhez és a költségtakarékos, mégis innovatív termék előállításához elvárt funkciókat.

A GigaDevice szabadalmaztatott „gFlash” memóriatechnológiával kiegészítve egy komoly mikro vezérlő vonal áll a tervezőmérnökök rendelkezésére. Az M3-család minden mikro vezérlője az ARM® Cortex®-M3 RISC processzormag köré szerveződik, mely a 108 MHz maximális órajelével és a beépített flash-memória azonnali elérhetőségével (Zero-Wait-State) maximális hatékonyságot biztosít.

A GD32® sorozatú mikrokontroller használata nemcsak a fejlesztők, de a felhasználók számára is sok előnnyel szolgál. Az MCU maximális sebessége a versenytársakénál 50%-kal többet nőtt. A kódfuttatás hatásfoka ugyanolyan órajel mellett 30-40%-kal nagyobb.

Az áramfogyasztás ugyanolyan frekvencia esetén 20–30%-kal csökkent. Ezek a tulajdonságai teszik lehetővé, hogy a GD32® sorozatú GigaDevice MCU-kat alkalmazások széles spektrumán lehessen használni.

A GD32 sorozatú mikrokontrollerek teszteléséhez és a fejlesztés

endrich

megkönnyítéséhez a GigaDevice különböző tudásszintű kiértékelő kártyákat és kezdőkészleteket kínál az egyszerű programozó és hibakereső moduloktól a maximális hardverkiépítésű teszt alaplapokig, ahogy azt cikksorozatunk előző részében részletesen tárgyaltuk.

GD32 Kezdőkészlet

A GigaDevice kezdőkészlet az MCU kivezetéseihez illeszkedő csatlakozó felületeket (Extension Header) kínál a felhasználó számára a gyors prototípus csatlakoztatáshoz és teszteléshez. Minden ilyen eszköz tartalmazza a GigaDevice saját GD-Link programozói és hibakereső interfészét is, melyen keresztül USB kábel segítségével kapcsolódhatunk a személyi számítógéphez, ezzel biztosítva a kártya tápellátását és az adatkapcsolatot is a mikrokontroller programozásához és a szoftverhibakereséshez.

1| GD32F170C8T6 GigaDevice GD32™ ARM® Cortex®-M3 kezdőkészlet

Fejlesztőeszközök - CrossWorks for ARM 4.1.

A GD32® család integrálja azokat az MCU jellemzőket, amelyek lehetővé teszik a gyors, könnyű és professzionális beágyazott rendszer-tervezést, és a fejlesztők kezébe ad egy megfizethető és bizonyítottan innovatív, komplex félvezető-gyártási technológián alapuló MCU eszközt. A GigaDevice számos jól ismert ARM fejlesztőrendszerhez kínál kiterjedt eszköztámogatást, így például a KEIL, az IAR vagy a „Rowley CrossWorks for ARM” platform-független integrált fejlesztői környezethez a mikrokontrollerek programozásához, hibakereséshez és ellenőrzéshez.

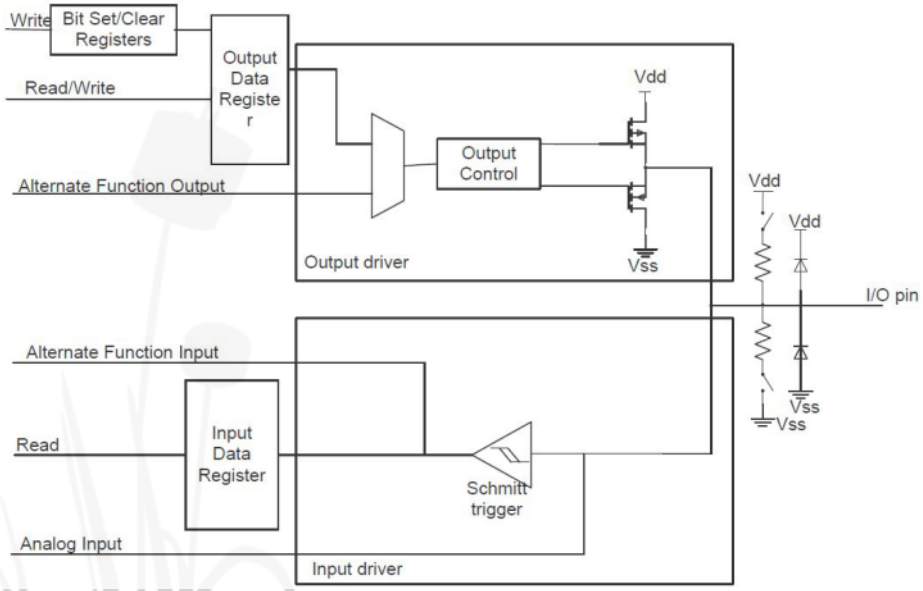
A népszerű ARM IDE a CrossWorks for ARM termékhez a gyártó speciális próba licencet ajánl, a felhasználó döntheti el, hogy (30 napos) időkorlátos teljes verziót, vagy 16 kB kódméretre korlátozott, egyébként teljes funkciós korlátlan ideig használható próbaváltozatot telepít. (A Keil MDK-ARM Lite Edition próbaváltozatként szintén rendelkezésre áll, itt 32 Kbyte a méretkorlát).

A CrossWorks for ARM egy komplett C/C++ és Assembly nyelvű fejlesztőrendszer, ami sok más mellett a Cortex-M mikrokontrollerekre való fejlesztést is messzemenőkig támogatja.

A CrossStudio integrált fejlesztői környezet egy natív módon felépített IDE, mellyel szerkeszthetjük, fordíthatjuk, a mikrokontroller Flash memóriájába tölthetjük a kódot és lehetőség van a hibakeresésre is az SWD/JTAG interfészen keresztül.

Általános célú I/O portok (GPIO)

A GD32F170C8T6 GigaDevice GD32™ ARM® Cortex®-M3 mikrokontrollerben 55 általános célú I/O port áll rendelkezésre, melyek 16-os blokkokba szerveződve a PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC15 lábakon, illetve a PD2, PF0/PF1, PF4 ~ PF7 lábakon érhetők el és biztosítanak a külvilág felé logikai kapcsolatot a hozzájuk rendelt vezérlő- és konfigurációs regisztereken keresztül. A GPIO portok által használt lábak megosztva más alternatív funkciókkal is rendelkezhetnek (AF - I²C, SPI, USART, CCP, PWM, Clock, ADC) és egyenként beállíthatók digitális kimenetként (kimenet választó regiszteren keresztül „push-pull" vagy „open-drain" módban), digitális bemenetként („pull up/down", vagy lebegtetett), valamelyik alternatív periféria funkcióra (pl. SPI MISO vagy MOSI) vagy analóg bemenetként (ADC) is. A portok maximális kommunikációs sebessége a kimeneti sebesség-regiszterek írásával változtatható, míg a „pull up/down" regiszterekkel kiválasztható, hogy a beépített „pull-up"

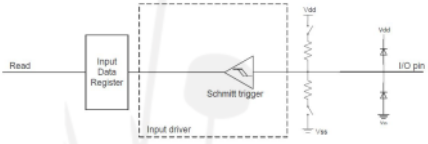


2| Az általános célú I/O bit felépítése

és „pull-down” ellenállások legyenek-e használva, amikor pl. közvetlenül egy kapcsolót kötünk az digitális bemenetre, vagy egyik sem, ha a GPIO-t lebegtetett módban kívánjuk használni (pl. külső felhúzó vagy lehúzó ellenállás alkalmazásakor). Ez utóbbi lebegtetett input mód az alapértelmezett beállítás, miközben az alternatív funkciók ki vannak kapcsolva. Az analóg bemeneti mód alkalmazása kivételével a GPIO portok nagy árammal terhelhetők.

Amikor a GPIO bemenetként konfigurált:

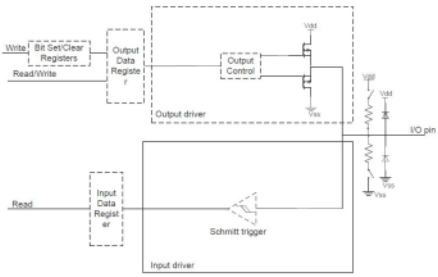
- A Schmitt Trigger bemenet aktivált
- A beépített pull-up és a pull-down ellenállások választhatók
- Minden AHB2 órajel ciklusban az I/O lábon megjelenő adat a bemeneti regiszterből kiolvasható (Data Input Register)
- A kimeneti puffer (Output Buffer) le van tiltva



3 | GPIO port bemenetként konfigurálva

Amikor a GPIO kimenetként konfigurált:

- A Schmitt Trigger bemenet aktivált
- A beépített pull-up és a pull-down ellenállások választhatók
- A kimeneti puffer (Output Buffer) engedélyezett:
  - Open Drain mód: A logikai “0” a kimeneti regiszterben aktiválja az N-MOS-t, míg a logikai “1” szint a portot nagyimpedanciás állapotba hozza.
  - Push-Pull mód: A logikai “0” a kimeneti regiszterben aktiválja az N-MOS-t, míg a logikai “1” a P-MOS-t aktiválja.
- Push-Pull módban a kimeneti adatregiszter olvasásakor az utolsó kiírt adathoz férünk hozzá
- A read access to the Data Input Register gets the I/O state in open drain mode

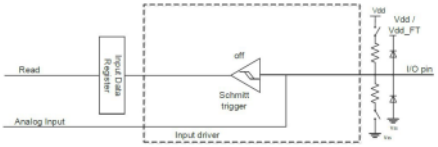


4 | GPIO port kimenetként konfigurálva

GPIO analóg konfigurációban:

- A beépített pull-up és a pull-down ellenállások kiválasztása le van tiltva
- A kimeneti puffer (Output Buffer) le van tiltva

- A Schmitt Trigger bemenete deaktivált
- A bemeneti adatregiszter olvasásakor „0” értéket kapunk.



5 | GPIO port ANALOG bemenetként

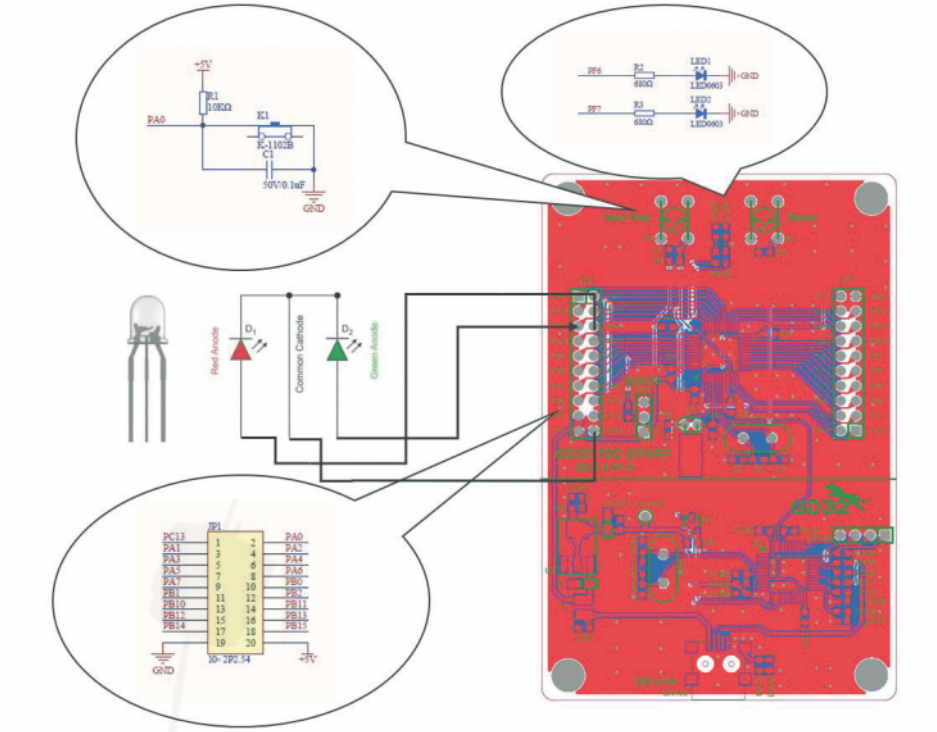
Analóg / Digital átalakítók (ADC)

A 12 bit felbontású A6D átalakító a fokozatos közelítés módszerét (successive approximation) használja az ADC lábon mért feszültségérték digitalizálására, két mintavétel között maximálisan 1µs idővel (sebesség=1MS/s). A GD32F170xx és felette a konverziós sebesség ennek a duplája, és minél kisebb felbontást választunk (10 vagy 6 bit) a mintavételi sebességet növelhetjük. Az A/D konverter 19 multiplexelt csatornája 16 külső és 3 kitüntetett belső forrás feszültségét mérheti. Ez utóbbiak a beépített hőmérséklet szenzor (NTC), a referencia feszültség és az elemfeszültség monitorozás céljára vannak fenntartva. Az ú.n. **Analog watchdog** funkció lehetővé teszi a felhasználó által definiált alsó és felső feszültség határértékekkel definiált tartományból való kilépés detektálását és kezelését egy automatikusan induló interrupt (IRQ) szoftveres feldolgozásával. Az A/D konverzió folyatható egyes csatornánként, vagy csoportosan folyamatos illetve szakaszos módon. A konverzió eredményét egy 16 bites regiszterben balra, illetve jobbra zárt módon kapjuk meg, vagy DMA segítségével időveszteség nélkül a processzormag megkerülésével közvetlenül a memóriába juttathatjuk a maximális mintavételi sebesség eléréséhez, hiszen ekkor nem kell annyit várni az előző eredmény feldolgozására az új mintavételhez. Az A/D konverter tápfeszültsége 2.6V - 3.6V, és így a közvetlenül mérhető feszültségtartomány $V_{SSA} \leq V_{IN} \leq V_{DDA}$.

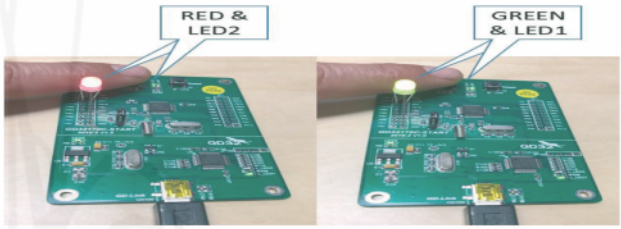
A lehetőségek bemutatása a kezdőkészlet használatával

A mikrokontroller GPIO-i és az A/D konverterei használatának bemutatásához egy egyszerű mintaprogramot készítettünk. A bemutatáshoz szükség van a kezdőkészlet két felhasználói LED-jére (LED1&LED2), melyek egyben a mikrokontroller PF6 és PF7 GPIO portjaihoz is kapcsolódnak, melyek az ábra jobb oldalán található csatlakozósoron is hozzáférhetők. Ide csatlakoztattuk egy kétszínű LED anódjait, míg közös katódját a GND kivezetéshez illesztettük. Ez a piros/zöld LED a kezdőkészlet LED1 és LED2 világító diódáival párhuzamosan működik majd.

A felhasználói interakció biztosítására a panel K1 felhasználói nyomógombját fogjuk használni, mely megnyomás esetén az 5V tápfeszültséget egy felhúzó ellenálláson keresztül a PA1 GPIO bemenetre kapcsolja. Az 5V megjelenése egy megszakítást (IRQ) generál, melyet a szoftverben kezelünk le, és használjuk fel a két LED alternatív be- és kikapcsolására. Többszöri gombnyomás esetén a LED1, a LED2, és a kétszínű LED piros illetve zöld chipje felváltva gyullad ki és alszik el.



6 |A GD32170C-START starter kit áramköri elemeinek magyarázata: LED1&LED2 – kezdőkészlet felhasználói LED-ek; K1- kezdőkészlet felhasználói kapcsoló; Kétszínű LED – a PF6 és PF7 GPIO és GND lábakra kötve



7 | GD32170C-START starter kit – Ismételt gombnyomásra a LED-ek felváltva működnek

A következő kódrészlet bemutatja a GPIO portok bemenetként (kapcsoló) és kimenetként (LED) való használatát.

```
//============================================================
// GigaDevice GD32170C-START kit demo
// Endrich GmbH - demonstartion program 2018.
//
// Bicolor LED's anodes are connected to the PF6 and PF7 pins
// PF6&PF7 are also connected to LED1 and LED2 of the board
// a key-button is connected to PA0.
//============================================================

#include <gd32f1x0.h>
#include <stdlib.h>
#include <stdio.h>


#define LED_PIN1 6
#define LED_PIN2 7
#define LED_PORT GPIOF
#define BUTTON_PIN 0
#define BUTTON_PORT GPIOA
#define BUTTON_EXTI 0


extern "C" void EXTI0_1_IRQHandler(void)
{
  static int count = 0;

  if (++count & 1)
    {
      // Turn LED on
      GPIO_BOP(LED_PORT) = 1 << LED_PIN1;
      GPIO_BC(LED_PORT) = 1 << LED_PIN2;
    }
  else
    {
      // Turn LED off
      GPIO_BC(LED_PORT) = 1 << LED_PIN1;
      GPIO_BOP(LED_PORT) = 1 << LED_PIN2;
    }

  // Clear interrupt
  EXTI_PD = 1 << BUTTON_EXTI;
}

int main(int argc, char *argv[])
```

```
int main(int argc, char *argv[])
{

  // Enable GPIOA and GPIOF clock
  RCU_AHBEN |= RCU_AHBEN_PAEN | RCU_AHBEN_PFEN;

  // Configure LED
  GPIO_CTL(LED_PORT) = (GPIO_CTL(LED_PORT) & ~(0x3 << (LED_PIN1 * 2))) | (0x1 << (LED_PIN1 * 2));
  GPIO_OMODE(LED_PORT) &= ~LED_PIN1;
  GPIO_OSPD(LED_PORT) |= 0x3 << (LED_PIN1 * 2);

  GPIO_CTL(LED_PORT) = (GPIO_CTL(LED_PORT) & ~(0x3 << (LED_PIN2 * 2))) | (0x1 << (LED_PIN2 * 2));
  GPIO_OMODE(LED_PORT) &= ~LED_PIN2;
  GPIO_OSPD(LED_PORT) |= 0x3 << (LED_PIN2 * 2);

  // Configure PA0 pin to generate EXTI0 interrupt on falling edge.
  GPIO_CTL(BUTTON_PORT) &= ~(0x3 << (BUTTON_PIN * 2));
  GPIO_PUD(BUTTON_PORT) &= ~(0x3 << (BUTTON_PIN * 2));
  GPIO_OSPD(BUTTON_PORT) |= 0x3 << (BUTTON_PIN * 2);
  SYSCFG_EXTISS0 &= ~7;
  EXTI_FTEN |= 1 << BUTTON_EXTI;
  EXTI_INTEN |= 1 << BUTTON_EXTI;

  // Set interrupt priority
  NVIC_SetPriority(EXTI0_1_IRQn, 1);

  // Enable interrupt
  NVIC_EnableIRQ(EXTI0_1_IRQn);

  // Endless loop
  while (1);
}
```

Az A/D konverter használatának bemutatásához a beépített NTC hőmérséklet szenzor szolgáltatta – hőmérséklettel arányos - elektronikus jel mintavételezését végezzük el. A termisztor ellenállása a hőmérséklet változásával ellentétesen alakul, így azt egy feszültségosztóban - egy precíz állandó ellenállás mellé kötve - felhasználhatjuk egy hőmérséklettel arányos feszültségérték szolgáltatására.

Ezt a feszültséget fogjuk az A/D konverter segítségével (annak kitüntetett ADC0 csatornáján keresztül) mérni.

A kódrészlet a következő (a kódot angol nyelvű magyarázatokkal láttam el):

```
/* ADC temperature sensor channel config */
adc_inserted_channel_config(ADC0,0,ADC_CHANNEL_16,ADC_SAMPLETIME_239POINT5);

/* ADC software trigger enable - Inserted channel A/D conversion */
adc_software_trigger_enable(ADC0, ADC_INSERTED_CHANNEL);
.
.
.
/* 2ms delay time before the measurement*/
delay_1ms(2000);

   /* value conversion: 12 bit data means 2^12=4096 possibilities
      therefore
                    0V represented by 0000
                    3.3V represented by 4095
      The ambient temperature is supposed to be 25C, for which the voltage
      is 1.42V. The NTC has an own R/T table so the voltage temperature change
      ratio can be expressed by 1000/4.3
   */
temperature = (1.42 - ADC_IDATA0(ADC0)*3.3/4096) * 1000 / 4.3 + 25;
```

A fenti mintaprogramok és a hozzájuk tartozó rövid magyarázatok a teljesség igénye nélkül ugyan, de alkalmasak a GigaDevice GD32™ ARM® Cortex®-M3/M4 mikrokontrollerek GPIO és ADC perifériáinak használatáról képet adni.

Bármely jól ismert beágyazott ARM fejlesztőrendszer (KEIL, IAR vagy CrossWorks IDE), és magas szintű programnyelv (C/C++) használatával szinte bármilyen feladatra alkalmazhatók ezek a kontrollerek.

Mivel a GigaDevice 32 Bit mikrokontroller funkcióiban, kivitelében nagyon hasonlít az ST / Freescale STM32 családjához, sokan váltanak manapság ezekre az eszközökre.

Az elektronikával hobbiszerűen foglalkozó szakemberek ma az Arduino világban használt IDE megtartásával már ARM alapú eszközöket is alkalmazhatnak és pl. STM32DUINO projekteken dolgozhatnak, ráadásul mostanra az online piactereken Gigadevice GD32 alapú “DUINO” modelleket is találhatnak.

A professzionális felhasználók számára azonban továbbra is javasoljuk a cikksorozatunkban bemutatott valamelyik kiértékelő vagy kezdőkészlet beszerzését, melyekkel ipari applikációk készíthetők, tesztelhetők és fejleszthetők.